

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática



MODELING CELL MIGRATION IN
QUANTITATIVE IMAGE ANALYSIS

Patrícia Andreia Cirne da Silva

Mestrado em Tecnologias de Informação aplicadas às
Ciências Biológicas e Médicas

2012

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática



MODELING CELL MIGRATION IN QUANTITATIVE IMAGE ANALYSIS

Patrícia Andreia Cirne da Silva

Trabalho orientado pelo Prof. Dr. André Osório e Cruz de Azerêdo Falcão
e co-orientado por Prof. Dr. António Alfredo Coelho Jacinto

Mestrado em Tecnologias de Informação aplicadas às
Ciências Biológicas e Médicas

2012

Abstract

All biological phenomena are dynamic and movement is an essential function in cellular systems but their regulation, characteristics and physiological meaning are not fully known. Measurement of the cell movements provides quantitative information that is inevitable for understanding the cellular system.

Cell migration is a field of intense current research generating high amounts of image data that need to be quantitatively analyzed with efficiency, consistency and completeness. To accomplish, computerized motion analysis is rapidly becoming a requisite. Since all the existing algorithms for these purposes are often not robust, effective and optimal enough to yield satisfactory results, new and alternative methods must be developed.

The aim of this work is to find and develop an alternative to the tracking of individual cells in order to, visualize, characterize and quantify the migration characteristics of cell population. This alternative comprises the implementation of a simple and automated algorithm to obtain qualitative and quantitative information from image sequences of cell migration in a fast, easy and inexpensive computationally way.

After an extensive literature review, it became clear that all the methodologies and approaches employed to make the quantitative analysis of cell migration only presented solutions that involved object tracking. And the new method developed estimates the probability density functions for cell migration and was implemented as a plugin (*Migration*) for ImageJ, as cross platform open source application. In the evaluation of the developed algorithm was taken in to account his applicability, efficiency, consistency, completeness and validity. It can be used to in image sequences to extract information regarding the distribution of the future positions of all particles in a determined time point in the future and is quick when is executing. The results obtained with this method were satisfactory.

Comparing to existing approaches to study the cell migration this method adds an improvement, it can deal with complex situation, such as overlapping of particles or other occlusions.

Keywords: Cell migration, ImageJ, particle tracking, plugin, quantitative image analysis.

Resumo

Todos os fenómenos biológicos são dinâmicos e o movimento é uma função essencial nos sistemas celulares, mas a sua regulação, características e significado fisiológico não são totalmente conhecidos. A medição dos movimentos das células providencia informação quantitativa para compreender o sistema celular.

A migração de células é um campo de intensa investigação gerando grandes quantidades de dados que necessitam de ser quantitativamente analisados com eficiência, consistência e de maneira completa. Para tal, a análise do movimento através dos sistemas de informação está a tornar-se cada vez mais num requisito. Dado que os algoritmos disponíveis para este propósito não são muitas vezes robustos, eficientes e óptimos para proporcionarem resultados satisfatórios, métodos alternativos devem ser desenvolvidos e implementados.

O objectivo deste trabalho é encontrar e desenvolver uma alternativa para o *tracking* de células de modo a se visualizar, caracterizar e quantificar a migração de células. Esta alternativa requer a implementação de um algoritmo simples e automático para obter a informação, quer qualitativa, quer quantitativa de um vídeo, com imagens da migração de células, de um modo rápido e fácil.

Depois de uma revisão bibliográfica extensa, verificou-se que todos os métodos implementados para fazer a análise quantitativa da migração de células eram soluções de *tracking* de partículas. O novo método aqui desenvolvido estima as funções de densidade de probabilidade para a migração de células e foi implementado como um plugin (*Migration*) para o ImageJ. A avaliação do algoritmo desenvolvido teve em conta a sua aplicabilidade, eficiência, consistência e validade. Pode ser usado em vídeos e extrair informação relativa à estimação da distribuição das posições de todas as partículas num determinado momento no tempo, executando de maneira rápida. Todos os resultados obtidos com este novo método são satisfatórios.

Comparando com as abordagens conhecidas da literatura, este método apresenta uma melhoria, pode lidar com situações complexas, tais como sobreposição de partículas e outras oclusões.

Palavras-Chave: Análise de imagem quantitativa, ImageJ, migração de células, plugin, *tracking* de partículas.

Acknowledgments

This work closes a four year cycle in my life, crowded with ups and downs, good and also bad moments that helped me understand life as it is. One of my accomplishments was never to give up when facing adversity and misfortune. So, I take this opportunity to show appreciation to all the people that helped me along this journey, giving me support, encouragement, relevance, concern and friendship.

I would like to express my sincere gratitude to my advisor Professor Dr. André Falcão and my co-advisor Prof. Dr. António Jacinto for allowing me the opportunity to develop this work.

To all my friends and colleagues - every moment shared together taught me to be a better person. To Pedro - his continuous support, encouragement, kindness and love keeps me going further.

My deeply gratitude to my parents, Rui and Anabela – they taught me that nothing is impossible when it comes to lend a hand to their children. Without them I would not be the person that I've become. To the rest of my family, I am thankful for the continuous support, encouragement and kindness.

Contents

List of Figures	ix
List of Tables	x
Chapter 1 Introduction.....	1
1.1 Context and Motivation.....	1
1.2 Goals.....	3
1.3 Methodology.....	3
1.4 Document Organization.....	4
Chapter 2 Theoretical background.....	5
2.1 Quantitative Analysis of Cell Migration	5
2.1.1 Particle Tracking and Data Analysis.....	6
2.2 Estimating Particle Positions in Image Sequences	8
2.3 Image J.....	9
2.3.1 Important Features	10
2.3.2 ImageJ Plugins	11
Chapter 3 Methodology and Implementation	14
3.1 Perception	14
3.2 Conception and Implementation.....	15
3.2.1 Target Audience	15
3.2.2 Algorithm: <i>Migration</i>	15
3.3 Evaluation.....	22
Chapter 4 Evaluation	25
4.1 Results Obtained.....	25
4.1.1 Manual Annotation vs. Automatic Annotation	26
4.1.2 Comparison with the wound location.....	31
4.2 Applicability and Efficiency.....	32
4.3 Consistency and Completeness	32

Chapter 5 Conclusions	34
References	37

List of Figures

Fig. 1. ImageJ main window (displayed on Windows XP).....	10
Fig. 2. ImageJ software structure (simplified)..	11
Fig. 3. Interface of the <i>Migration</i> plugin for ImageJ.	18
Fig. 4. Interface of the <i>Particle Detector and Tracker</i>	18
Fig. 5. Interface of the <i>Migration</i> plugin.....	18
Fig. 6. Interface of the <i>Migration</i> plugin for ImageJ.	19
Fig. 7. Representativity map obtained with the <i>Migration</i> plugin.	19
Fig. 8. Permanence probability map obtained with the <i>Migration</i> plugin.	20
Fig. 9. Results obtained with the <i>Migration</i> plugin	21
Fig. 11 Influence of p in the outcome of the <i>Migration</i> plugin.....	28
Fig. 12 Influence of a in the outcome of the <i>Migration</i> plugin.....	29

List of Tables

Table 1. Images used in the evaluation of the <i>Migration</i> plugin and their characteristics.....	22
Table 2. Parameters needed to run the <i>Particle Detector and Tracker</i> plugin.....	23
Table 3. Range of values defined by user in the iterations of the <i>Migration</i> plugin.....	24
Table 4. Best results of the <i>Migration</i> plugin from the image sequence ;SrpHemoGal4UASnlsCherryUbiCad;after_proj.tiff (FilmB) – Manual Annotation.....	29
Table 5. Best results of the <i>Migration</i> plugin from the image sequence ;SrpHemoGal4UASnlsCherryUbiCad;after_proj.tiff (FilmB) – Automatic Annotation.....	30
Table 6. Best results of the <i>Migration</i> plugin from the image sequence SrpHemoGal4UASnlsCherryUbiCadGFPaft_proj.tiff (Film A)) – Manual Annotation.....	30
Table 7. Best results of the <i>Migration</i> plugin from the image sequence SrpHemoGal4UASnlsCherryUbiCadGFPaft_proj.tiff (Film A)) – Automatic Annotation.....	31
Table 8. Coordinates and area of the wound location.....	31
Table 9. Distances of the mean values to the center of mass of the wound.....	32

Chapter 1

Introduction

1.1 Context and Motivation

All biological phenomena are dynamic and movement is an essential function in cellular systems. As examples: vesicles move to specific sites within cell for their destined functions; chromosomes move to separate from each other during cell division; cytoskeleton dynamically forms bundles and networks to afford routes for the intracellular trafficking and to control the physical architecture of the cell; and, cells themselves move by migration. Within multi-cellular structure, cell movement drives the morphogenesis and preserves the homeostasis but their regulation, characteristics and physiological meaning are not fully known. Measurement of these various movements provides quantitative information that is inevitable for understanding the cellular system [16,19,20,23,24].

Cell migration, either randomly or directionally, is a highly dynamic phenomenon and an important biological parameter in many different biological processes and contexts (e.g. morphogenesis, wound healing, cancer metastasis and immune response) [2,6,18,29,31]. A field of intense current research, the understanding of all mechanisms of cell migration has become one central goal for scientists and researchers, because unregulated migration can lead to the emergence of diseases [1-3,5-7,11,12,15,16,18-20,23-31].

All studies, in biomolecular dynamics in general and in cell migration in particular, generate substantial amounts of image data. In this case, even the qualitative analysis becomes difficult to do, since only the most drastic alterations of motility patterns can be perceived by the eye [16,23,20,28]. These data needs to be analyzed quantitatively and as completely as possible to reveal functional behavior and to detect rare and subtle phenomena, allowing the design of accurate mathematical models of

dynamic structures and cell movement. To accomplish this, computerized motion analysis is rapidly becoming a requisite [7,16,19,20,23].

Also, with the standard systems or with the development of new methods of imaging in microscopy, it becomes important to make available new techniques for quantitative image analysis in order to track and measure the movement of cells or organelles [2,23-25,27]. According to Sbalzarini & Koumoutsakos, techniques such as multi-color video microscopy and particle tracking are becoming indispensable in cell biology, where quantitative analysis of the trajectories provides important information about working mechanisms and structures in living cells [27].

There are three important motivations for applying computerized techniques in quantitative image analysis for cell migration:

- ◆ Efficiency: With the high amounts of image data generated from biomedical experiments, it becomes challenging the management of those data and unrealistically executable the manual analysis. To overcome these issues, there must be an efficient and automated extraction of quantitative measurements [1,3,15,18,20,23,26-28,31].
- ◆ Consistency: computational image analysis generates consistent data, i.e. different experiments are processed based on the same parameter settings and criteria for the validation of measurements. This aims to expunge uncertainty associated with subjective interpretations among researchers and even by the same researcher in distinct instances [1,3,9,15,16,18,20,23,26,28,30,31].
- ◆ Completeness: complete data is also produced by computational image analysis. In manual analysis, the researcher has the tendency of being concentrated on the apparently interesting events, adding bias/error to the analysis. In contrast, for computational analysis, every image event that fulfills an objective set of criteria is considered in the analysis [9,15,18-20,23,28,30].

Kaladztis stated that: a good, accurate and robust algorithm should deliver data set that provides statistically reliable and non-biased results [9,19,20]. Since all the existing algorithms for this purpose are often not robust, effective and optimal enough to yield satisfactory results [3,15,26,28], new and alternative methods (algorithms) to analyze the data sets of cell migration must be implemented and developed.

1.2 Goals

The aim of this work is to attain, learn and develop an alternative method to particle tracking of cells in order to visualize and characterize the movement characteristics of cell population. This alternative comprises the development and implementation of a simple and automated algorithm to obtain qualitative and quantitative information from image sequences of cell migration in a fast, easy and inexpensive computationally way.

The specific goals for the completion of this task are the following:

- ◆ Understand the problem at hand through an extensive literature review;
- ◆ Draw the alternative solution (algorithm) to the techniques already implemented for tracking cells;
- ◆ Implement the algorithm as a plugin (*Migration*) for the image analysis software ImageJ;
- ◆ Test and evaluate the implemented plugin and his applicability in a set of image sequences.

1.3 Methodology

To meet the established objectives, the following methodology was used for this work:

- ◆ Perception – this stage includes a review made to the available literature, in order to fully understand the problem at hand and acknowledge the solutions developed by other researchers.
- ◆ Conception – in this phase was drawn a new and alternative method to particle tracking, that also allows the migration study of cells in image sequences.
- ◆ Implementation – the algorithm developed in the last stage is implemented as a plugin for ImageJ using the Java programming language.
- ◆ Evaluation - the algorithm is tested in image sequence, in order to ascertain his applicability, efficiency, consistency, completeness and validity.

All this four phases/stages are interdependent and complementary. The features of the *Migration* plugin and his interface must be suitable to the tasks and be capable to

solve the problem and deliver the desired results. This means that, if the first solution does not deliver de appropriate results another solution must be conceived, implemented and tested.

1.4 Document Organization

This document is organized in five chapters, as follows:

- ◆ Chapter 1 – frames and presents the problem under study as well as the motivation and the goals to find and implement a solution for it.
- ◆ Chapter 2 – introduces some concepts about quantitative analysis in cell migration necessary to contextualize this work.
- ◆ Chapter 3 – describes how the algorithm works and how it was implemented as a plugin for ImageJ.
- ◆ Chapter 4 – presents and discusses the results obtained, that allow the evaluation of the implemented plugin *Migration* and its interface.
- ◆ Chapter 5 – sets out the main conclusions of this work and the future directions for new researches.

It is very important to mention that the words “cell”, “particle” and “object” will be used interchangeably and refer to the entities to be analyzed over time in the image sequences.

Chapter 2

Theoretical background

2.1 Quantitative Analysis of Cell Migration

Dynamic sequences are used to study the active behavior of particles in a living organism (cell). Dynamic image is what usually is called “video sequences” or simply “movies”. The images are acquired sequentially in time at a suitable rate, since movement in biology inevitably deals with dynamic processes that change with time [12,24,26]. This offers an enormous potential for increasing our understanding of biological events, but it also constitute a challenge for quantitative analysis, which demands efficient techniques to evaluate this unparalleled flow of data [26].

The quantitative analysis of cell migration has several important functions: it gives a concise and accurate description of the process and can detect subtle differences in motility due to different experimental conditions; results can be communicated unambiguously to test hypotheses about motility; and, finally, it may be used to derive a mathematical model and thus to elucidate the underlying mechanisms [5]. Different parameters, whose significance may depend on the goals and interests of the biomedical research experiments, characterizes cell motility and dynamic properties: number of particles, position, spatial distribution, movement phases, speed, migration angle, diffusion coefficients, mean displacement, among others [5,7,15,19,20].

To perform reliable quantitative analysis of the particles behavior is necessary the detection and tracking of large and time-varying numbers of nano-metric scale objects in the image sequence. So, at each time point the location (coordinates) of all particles has been estimated [7,15,19,28].

2.1.1 Particle Tracking and Data Analysis

The goal of tracking particles is to extract clues about their dynamics and to make inferences about the laws of motion and/or unknown modeling parameters [12]. The tracking problem consists of two stages: detecting particles in a digital video sequence and linking these detections over time to follow the trajectories of individual particles [11,12,16,19,20,23,24,27,28,30].

There have been numerous applications of the tracking problem in several fields of science and technology, such as fluid mechanics [21], computer vision [8], navigation [17], material science [4], medicine and biology [1,3,15,16,18,20,25-28,30,31], among others.

Nowadays, are available innumerous specialized algorithms and computer programs, which are very often specific for a given application (the algorithm developed for one application tend to be sub-optimal or even useless in other applications [19,20]). For most of the implementations, is used a priori knowledge about the model of the movement or about the physics of the problem to construct effective, optimal and robust tracking procedures. In biological applications, the tracking is, very often, made in objects whose type of motion is not quite known explicitly in advance, by the possible stochastic character of the motion, or by trajectories entailing several modes of motion [19,20,27,30] (see [19,20] for more examples).

It is worth mentioning that the number of publications on cellular and intracellular object tracking algorithms is less than in other fields of science and technology and the algorithms are less sophisticated [20].

2.1.1.1 Manual analysis

One of the most common and simplest approaches used to provide data about the kinetics and dynamics of cellular and intracellular interactions is the manual analysis. In innumerous references is also called manual tracking of particles [19,20,26,30].

The positions of the particles are marked across consecutive frames in an image sequence, thus following (tracking) positional changes of the particles over time. This tracking procedure is commonly performed manually through “point and click” systems [18] (e.g. the *Manual Tracking* plugin for ImageJ).

Besides being time-consuming and labor-intensive, manual analysis is costly, inaccurate, poorly reproducible, highly dependent on operator’s skills and perception,

and usually only a small fraction of the data can be analyzed in this manner, leading to a loss of important information [3,5,18-20,23,26,28,30].

This methodology is susceptible to bias/errors regarding the unconscious selection of representative particles for analysis which satisfy the researcher's non-formalized criteria of "good data" [18-20,28]. Genovesio A et al asserts that many biological object motility studies are based on the study of a few hand-picked particles which represent only a small subset of the total (rarely all particles in an image sequence are considered) [15].

Countless authors stated that all these problems could be overcome by the automated tracking and analysis of particles [3,15,16,18-20,23,28,30].

2.1.1.2 Automated Analysis

For more than two decades, several methods have been developed to record the movement of cells through automatic methodologies. Also, referred to as automatic analysis or automatic tracking [1,3,15,18-20,26-28,31].

The majority of approaches for tracking particles in bioimaging consists of two distinguishable tasks/steps: particle identification on every frame of the image sequence (object/feature point detection or segmentation) and connecting identified objects in sequential frames into a chain, which belongs to the same physical entity (tracking or trajectory linking) [11,16,20,23,24,26-28,30]. Since these two procedures can be considered independent in most approaches, without the possibility of feedback from linking to detection and vice versa, the tracking performance of such methods is often suboptimal and extremely sensitive to failures in either stage [19,28]. It is important to notice that the performance of the tracking algorithm (rather than the segmentation algorithm) defines the fundamental performance limit of the method [11].

In order to be generally applicable, nearly every approaches are frequently based on rudimentary detection algorithms (thresholding [16,19,20,23,24]; gaussian fitting [16,19,20,23,24] or template/pattern matching [19,20,23,24]) and linking strategies (nearest neighbor [19,23,24]; graph theory [19,24] or smooth motion [20]), bringing limitations to the wider use in bioimaging applications [11,15,19,20,23,24,26,28]. Even if the algorithms are used in biomedical experiment (according to Cheezum MK et al, many laboratories develop custom-written software for analyzing the data) and

incorporate additional thresholds and filters to improve the consistency of their results, they are near from being rudimentary [11].

One of the unusual characteristics of most past and current approaches to particle tracking, however, is the rather strict discrimination between spatial and temporal information. Methods for detecting particles and estimating their positions per frame typically rely on spatial image features only and do not incorporate information from other frames. Nor do they yield many clues regarding possible inter-frame correspondences [23].

Only a few algorithms implementations make use of information from the tracking procedure to guide object detection, or in other words, there is an interaction between object detection and linking, but the execution remains far from what is required [19,20,28].

2.2 Estimating Particle Positions in Image Sequences

In order to overcome all the issues and problems of particle tracking, other solutions have been developed. Those solutions have the main goal to predict the future location of the object position.

Motion prediction is an investigation area with relevance in many diverse domains. Many research efforts on motion prediction found in the literature are based upon an a priori motion model, that portrays how the state of an object (e.g. position or velocity), changes over time when it is subject to a given control (e.g. acceleration) [32,33].

In order to predict the future motion of an object, its current state and control are estimated first. Then, the estimated state and control are fed into the object motion model in order to get future state estimations. Provided that the motion model used is correct and that the state and control estimations are accurate, such methods calculate good motion predictions. Unfortunately, these circumstances are rarely encountered and this kind of procedures is suited for short term motion prediction only [32].

To overcome these issues, different approaches had been developed. For a given area, moving objects have a tendency to follow typical motion patterns that depend on the objects' nature and the structure of the environment [32,33].

It operates in two stages. The learning stage, to observe the moving objects in the workspace in order to determine the typical motion patterns. And the prediction stage,

to use the learned characteristic motion patterns to predict the future motion of a given object [32].

This allows the classification of the techniques following this approach as: grid-based techniques; derived from the occupancy grid concept. The environment is demonstrated as a grid and the learning stage calculates the transition probability for a moving object from one grid cell to another (the grid is used directly for motion prediction. The other classification is cluster-based techniques: sets of partially or wholly similar observed trajectories are clustered together. A representative trajectory for each cluster is computed. Such representative trajectories are used for motion prediction, since they permit to take into account not only the current state of the object but also its past states. Cluster-based techniques are by far the best ones when it comes to long term motion prediction. Their only weakness lies in their inability to predict atypical trajectories [32,33].

2.3 Image J

ImageJ is a public domain image processing program implemented entirely in Java. Is fundamentally platform-independent, running, either as an online applet or as a downloadable application, without modification under Windows, MacOS or Linux [10,14].

It offers a set of ready-made tools for viewing and interactive manipulation of images, but can also be extended easily by writing new software components in Java programming language. The open architecture of ImageJ allows new modules (“plugins”) to be written as independent pieces of Java code that can be compiled, loaded, and executed in the running system without the need to even restart ImageJ (for this task, the software possesses an editor and a Java compiler). This attribute makes ImageJ an ideal platform for developing and testing new image-processing techniques and algorithms [10].

Being public domain open source software, an ImageJ user has the four essential freedoms: run the program, for any purpose; study how the program works, and change it to make it do what is wished; redistribute copies so it can help others; improve the program, and release the improvements to the public, so that the whole community benefits [14].

Besides being a great tool, ImageJ is naturally not perfect. From a software engineering point of view, its architectural design does not always seem intuitive and lacks of orthogonality, i.e. several tasks could be accomplished in a variety of different ways [10].

The actual version of ImageJ (v1.44p), updates, documentation, source code, test images and plugins can be found and downloaded from the ImageJ website (“<http://rsbweb.nih.gov/ij/>”).

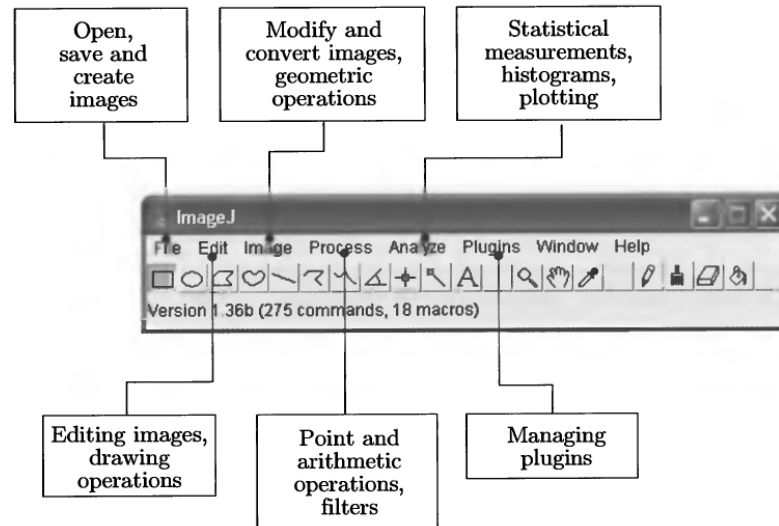


Fig. 1. ImageJ main window (displayed on Windows XP). Source: Burger W & Burge MJ, 2008 [10].

2.3.1 Important Features

As a Java application, ImageJ runs on any computer that has installed the current Java Runtime Environment (JRE). It can also be used as an applet within a web browser [10].

The most important features for this platform are [10,14]:

- ◆ A set of interactive tools for creating, displaying, editing, analyzing, processing, loading, saving and printing images (8-bit, 16-bit and 32-bit), with the support for several common file formats (.tiff, .gif, .jpeg, .bmp, .png, .dicom, .fits and “raw”).
- ◆ A simple plugin mechanism for extending the functionality of ImageJ by writing pieces of Java code.

- ◆ A macro language and the corresponding interpreter, which make it easy to implement larger processing blocks by combining existing functions without any knowledge of Java.

The program supports any number of windows (images) simultaneously, limited only by available memory. Spatial calibration is available to provide real word dimensional measurements in units such as millimeters [14].

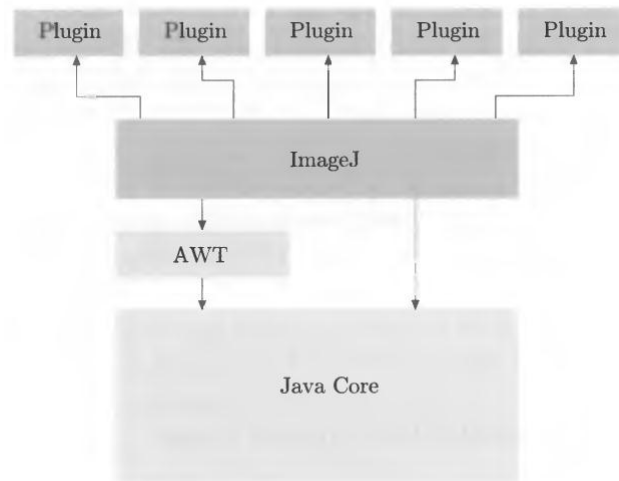


Fig. 2. ImageJ software structure (simplified). Source: Burger W & Burge MJ, 2008 [10].

2.3.2 ImageJ Plugins

Plugins are Java code loadable modules for extending the functionality of the basic ImageJ by using a simple standardized interface. ImageJ is based on the Java core system and depends in particular upon Java's Advanced Windowing Toolkit (AWT) for the implementation of the user interface and the presentation of image data (Fig. 2) [10,14].

Plugins can be created, edited, compiled, invoked, and organized through the Plugin menu in ImageJ's main window (Fig. 1) [10].

Technically, plugins are Java classes that implement a particular interface specification defined by ImageJ. This means that all features of the Java language can be used, the full ImageJ Application Programming Interface (API) can be accessed and all Java API can also be used. There are two different kinds of plugins [10,14]:

- ◆ PlugIn: no image is required when starting the plugin.

- ◆ **PlugInFilter**: the currently active image is passed to the plugin when started.

The first type can add support for new file formats and the latter can filter or analyze images [14].

2.2.2.1 *Manual Tracking* [13]

This plugin is accessible for use in ImageJ and was developed by Fabrice P. Cordlières (Institut Curie, Orsay, France).

It allows the quantification of objects movement between frames in an image sequence or temporal stack, in 2D and 3D. This plugin provides a way to retrieve in a table XY and XYZ coordinates as well as velocity, distance covered between two frames (not necessarily consecutive) and intensity of the selected pixel or volume, by simply clicking on the structure of interest. As explained earlier in the text, a simple “point and click” system [18].

This plugin also allows the possibility of visualize the detected particles and their respective trajectories, overlapping them on the original image sequence or in a new black image.

2.2.2.2 *Particle Detector and Tracker* [22]

This plugin, also available for ImageJ, was designed by Sbalzarini & Koumoutsakos (Mosaic Group, Computational Biophysics Lab, ETH Zurich, Switzerland).

It presents an easy-to-use, computationally efficient, two-dimensional, feature point tracking tool for the automated detection and analysis of particle trajectories as recorded by video imaging in cell biology. The tracking process requires no *a priori* mathematical modeling of the motion, it is self-initializing, discriminates spurious detections, and it can handle temporary occlusion as well as particle appearance and disappearance from the image region.

The plugin is well suited for video imaging in cell biology relying on low-intensity of fluorescence microscopy. The algorithm is fast and efficient, while at the same time having accuracy and precision that are comparable to far more computationally intensive algorithms [27].

Chapter 3

Methodology and Implementation

3.1 Perception

Comprehensive and automated analysis of large scale experimental data is an urgent item on the biomedical research agenda, placing image analysis into the center of progress [23]. After an extensive literature review, it became clear that all the methodologies and approaches employed to make the quantitative analysis of cell migration only presented solutions that involved object/particle tracking.

As it was mentioned in Chapter 1, the approaches of object tracking are very often not robust, effective and optimal enough [3,15,26,28]. So a new and alternative approach must be shaped to examine the data of cell movement.

In biological fields, the particle tracking algorithms are used to examine objects whose type of motion is, in most cases, unknown [19,20,27,30]. In addition to this, biological particles can entail several modes of movement, which some may have stochastic character [19,20]. In the approaches for particle tracking, the procedures of detecting (segmentation) and tracking (trajectory linking) are independent from each other. As Cheezum said, the performance of the tracking procedure defines the fundamental performance limit of the method [11]. The lack of feedback from one stage to another renders failures in the applications to analyze data, and consequently the tracking performance is far from desirable. In most cases, slight changes in detection may lead to the linkage of two different particles in the same trajectory [11,16,19,20,23,24,26-28,30].

One of the unusual characteristics of most past and current approaches to particle tracking, however, is the rather strict discrimination between spatial and temporal information. Methods for estimating particle positions along a video sequence typically do not incorporate information from other frames, they just rely on spatial frame features [23].

The new alternative technique to the tracking methods should be designed taking into account all the issues presented (no need to have knowledge about the motion, overcoming the limitations of particle tracking and incorporating temporal or spatial information from other frames) in order to achieve efficiency, consistency, completeness, ease of use, availability, scalability and automation. Ultimately, must deliver statistically reliable and non-biased results. In addition, the algorithm and its implementation must be simple, fast and computationally inexpensive.

3.2 Conception and Implementation

3.2.1 Target Audience

The target audience that will benefit from the developed algorithm *Migration* is essentially researchers that develop their investigation in cell migration. Other users may exploit the plugin, such as professors or students in the biomedical field. However, being implemented as a plugin for ImageJ, a public domain image processing program, will grant access to everybody and allow each one to explore and use the plugin.

3.2.2 Algorithm: *Migration*

3.2.2.1 Theory

As explained previously, in order to achieve efficiency, consistency and completeness, the alternative solution to the tracking methods should be designed taking into account several issues: no need to have knowledge about the motion; overcoming the limitations of particle tracking; and, incorporating temporal and spatial information from other frames.

First of all, for the new approach it is not necessary to have knowledge about the motion undertaken by the particles because the method will assume that the velocity will be constant (the information used to compute the velocity is extrapolated from two consecutive frames). In this case, it is proposed to do a linear projection to predict a future particle position.

The limitation of particle tracking is completely overcome. In this new method there will be no tracking, only the prediction and estimation of the location of the particle thro time. The linear projection is done from each time frame to a time point in

the future that can be the same as the end of the image sequence or other time even further in the future.

Other thing that was taken into account was the incorporation of temporal and spatial information from other frames. The knowledge of the particle position in two consecutive frames, enables the computation of velocity between frames. The estimation of the particle position (linear projection) is applied to distinct differences of time, from each frame to a given time point (selected by the user). A value of uncertainty is associated to the prediction of the particle position to help discriminate the values with higher weight.

The plugin *Migration_* was, also, developed taking into account the following notions regarding the usability: being capable to use data that has to be immediately extracted from an image sequence and being capable to handle information, saved in a text file, formerly extracted from an image sequence. The first standpoint takes into account the need to have a fast, handy and automatic way to analyze images that have never been studied. The latter, features the employment of data that could be extracted from previous experiments, granting some flexibility in the manner of acquiring the material for analysis, by automatic or manual techniques.

The following presentation of the algorithm will resume everything:

Estimation of probability density functions for cell migration:

Given P_{it} as the (x,y) position coordinates of a particle i in frame t and being possible to follow that same particle from frame t to frame $t+I$, is feasible to determine its velocity vector V_{it} . Assuming a constant velocity, it is possible to make a linear projection of where the particle might be at a certain time point u :

$$P_{iu} = P_{it} + (u - t) V_{it}$$

However, smaller distances from t to u , endorses higher assertion that the particle will be in the position i predicted. Ultimately, if $u = t+I$ the confidence is total. Thus, it is used an attribute Q for the uncertainty and dependent on $d = u-t$:

$$Q_{tu} = d^{-a}$$

where a is a value set by the user. As higher the value a , larger will be the increase of the uncertainty associated with a time difference. For values of $a = 0$, the uncertainty is eliminated.

Thus, for each particle at a given time point u , we will have the position and a value associated with the uncertainty. The aim is, then, to predict, for each particle, in a given time point, the position and uncertainty.

3.2.2.2 Practical Resolution

The practical resolution must address the issues related to the ease of use, availability, scalability and automation. In addition, the implementation of the algorithm must be simple, fast and computationally inexpensive.

The existence of ImageJ allows a simple implementation, not having to develop new image processing software. The open architecture allows new modules written in Java that can be used (compiled, loaded, and executed) in the running system without restarting the software. This attribute makes ImageJ an ideal platform for developing and testing new image-processing techniques and algorithms [10].

Migration Plugin:

The automatic analysis plugin (*Migration*) was implemented in ImageJ using Java and consists of a graphical, cross platform open source application.

The plugin is prepared to run in two distinctive modes, corresponding to the availability of the data, i.e., to run with uploaded data files of particles positions previously obtained with other methods or to run calling the *Particle Detector and Tracker* plugin to automatically detect the particles positions used to estimate the probability density functions for the migration of cells.

The algorithm developed consists of the following steps:

- i) The algorithm asks the user for input data or parameter settings required to execute the plugin.

Using the *Particle Detector and Tracker* with the *Migration* plugin [22]
(Fig.3, 4 and 5):

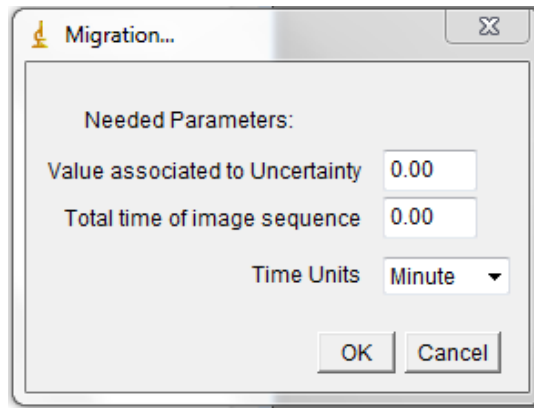


Fig. 3. Interface of the *Migration* plugin for ImageJ.

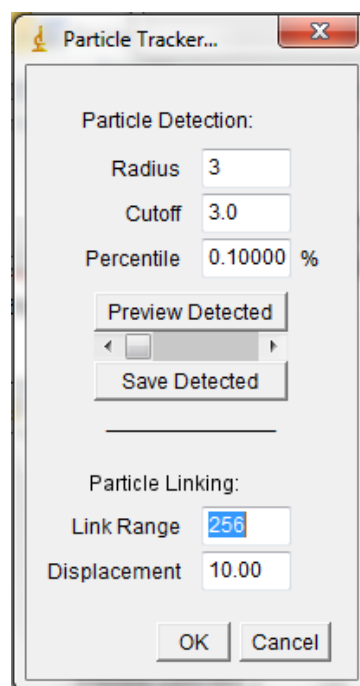


Fig. 4. Interface of the *Particle Detector and Tracker* (is called after the OK button was pressed in the last window from the *Migration* plugin).

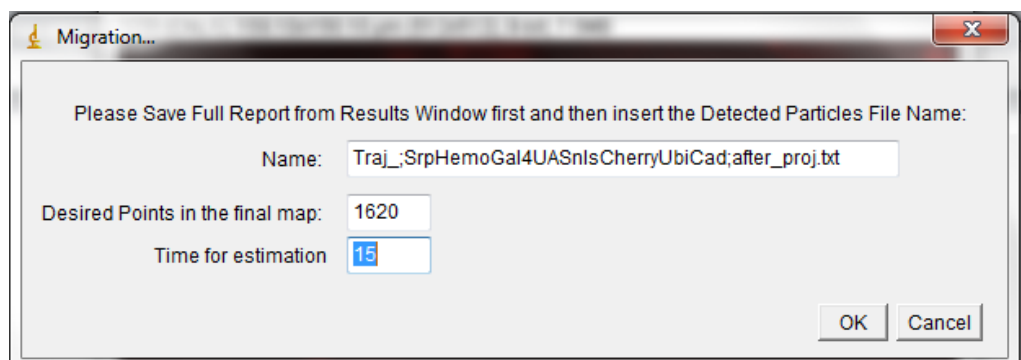


Fig. 5. Interface of the *Migration* (called after the OK button was pressed in the last window from the *Particle Detector and Tracker* plugin).

Using the data file obtained in the *Manual Tracking* or other suited file with information regarding the position of particles [13] (Fig.6):

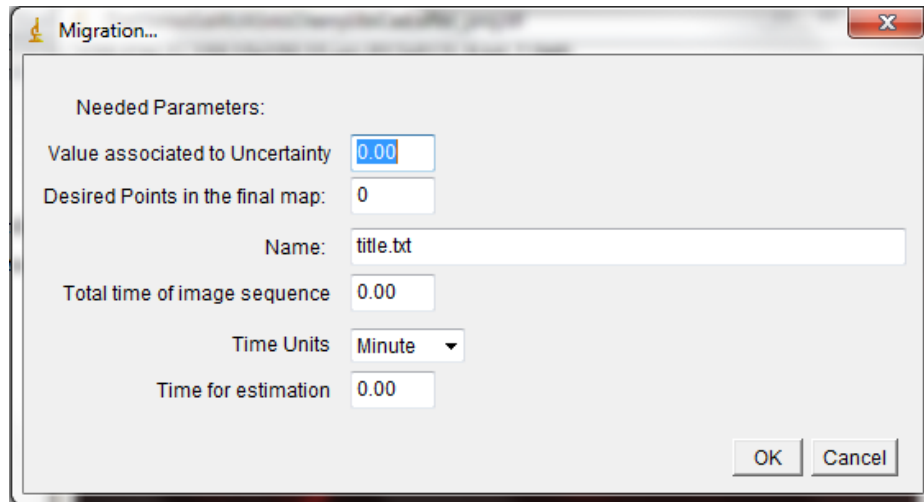


Fig. 6. Interface of the *Migration* plugin for ImageJ.

- ii) track each particle from one frame to another (each pair of frames) and determine the associated speed vector, as well as the value Q associated to the frame;
- iii) determine the positions P_{iu} of each particle;
- iv) do the representativity map of each pair particle/frame (Fig.7) using value Q for the uncertainty;

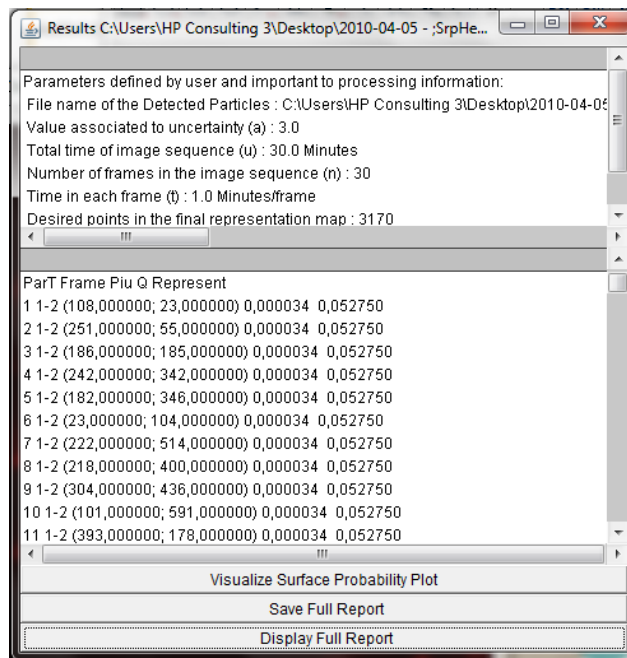


Fig. 7. Representativity map obtained with the *Migration* plugin (this window allows the user to visualize the probability plot, save the report results on a text file mode or display in the window the report).

- v) create permanence probability map, from the representativity map, over-proportionally representing the smallest uncertainties (ensure that all are adequately represented, according to a number of particles that have been defined in a final map (this number is also set by the user).
- vi) Create a data file: for each particle representativity, save as many lines as necessary (Fig.8).

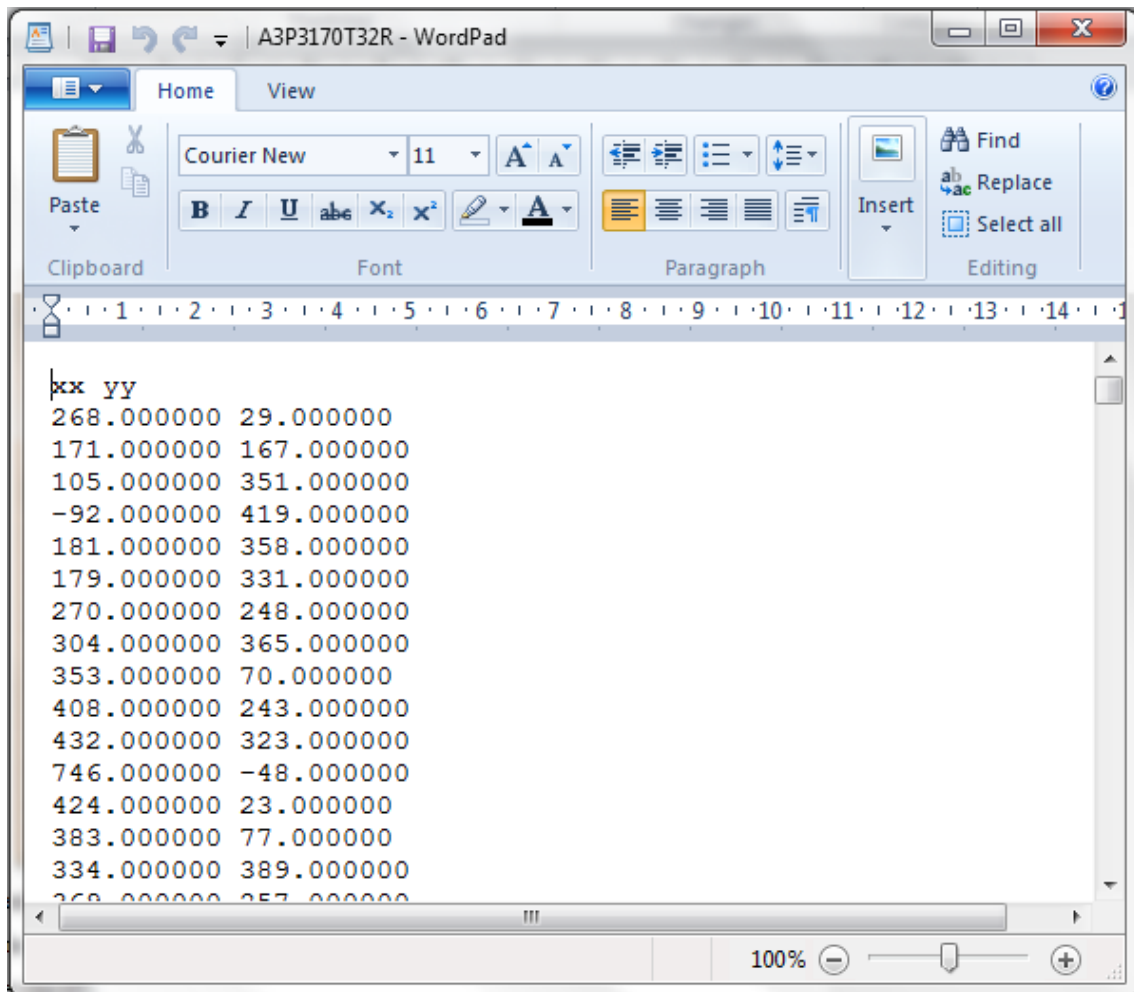
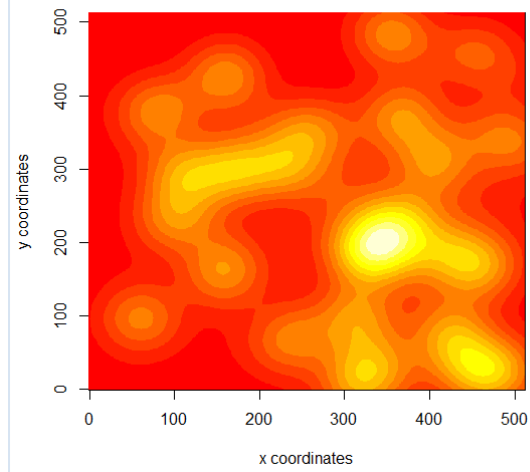


Fig. 8. Permanence probability map obtained with the *Migration* plugin.

- vii) Using a statistical tool (R), generate probability surfaces according to a simple kernel function (kde2d(), from the MASS library).
- viii) Visualize the surfaces with the functions (persp(), contour() or image()).



a)

3.3 Evaluation

As said previously, the plugin *Migration_* was developed taking into account two notions: being capable to use data that has to be immediately extracted from an image sequence and being capable to handle information, saved in a text file, formerly extracted from an image sequence. The first standpoint takes into account the need to have a fast, handy and automatic way to analyse images that have never been studied. The latter, features the employment of data that could be extracted from previous experiments, granting some flexibility in the manner of acquiring the material for analysis, by automatic or manual techniques.

In order to evaluate the *Migration_* plugin, the following method was employed:

Resources:

- ◆ Two images sequences of the *Drosophila Melanogaster*, featuring the hemocyte behavior, obtained by Confocal Microscopy. This images sequences were recorded shortly after inflicting a wound in the epithelium of the fly.

Table 1. Images used in the evaluation of the *Migration* plugin and their characteristics.

<i>Name</i>	SrpHemoGal4UASnlsCherryUbiCadGFPaft_proj.tiff (Film A)	;SrpHemoGal4UASnlsCherryUbiCad;after_proj.tiff (Film B)
<i>Dimension</i>	512 X 512 pixels	512 X 512 pixels
<i>Number of frames</i>	15	30
<i>Time interval</i>	1 frame/minute	1 frame/minute
<i>Size</i>	3,75 MB	7,50 MB

- ◆ Software:
 - ImageJ v1.44p (to run the plugin *Migration_.java*);
 - R (the statistical package to perform the statistical analysis);

Methods:

- ◆ First of all, the two images were annotated manually with the *Manual Tracking* plugin on ImageJ, in order to get the particles positions in all frames (this procedure was performed two times by two different individuals) saved in a text file;

- ◆ The data files obtained in the previous step were then used with the *Migration* plugin in numerous iterations (the values used for the parameters carried different combinations in each iteration, see table 3);
- ◆ Then, the *Migration* plugin was employed directly in both images, also in several iterations;
 - Since the developed algorithm calls the running methods of the *Particle Detector and Tracker* plugin to automatically detect and record all particle positions along the image sequence, the values of the parameters needed to run the latter plugin were kept the same for each use of the *Migration* plugin (the values used appear by default in the plugin, see table 2).

Table 2. Parameters needed to run the *Particle Detector and Tracker* plugin (the values used appear by default in the plugin).

<i>Parameters</i>	<i>Values</i>
Radius	3 pixels
Cutoff	3
Percentile	0,1
Displacement	10
Link Range	256

- The other required parameters, important to run the *Migration* plugin, also took distinct sorts of combinations in each iteration (see Table 3).
- ◆ At the end, the statistical analysis was performed using the software R (<http://www.r-project.org>).
 - For each iteration of the *Migration* plugin, it was calculated the mean and the variances of the estimated points.
- ◆ The ImageJ was used once again to determine the x and y coordinates of the center of the wound inflicted in the fly.

The results obtained with evaluation done to the *Migration* plugin are presented in the next chapter.

Table 3. Range of values defined by user in the iterations of the *Migration* plugin.

<i>Variables</i>	<i>Values</i>					
	Manual Annotations				Automatic Annotation	
	Manual A		Manual B		Automatic Annotation	
	Film A	Film B	Film A	Film B	Film A	Film B
Value associated to uncertainty (α)	[1; 5]	[1; 5]	[1; 5]	[1; 5]	[1; 2]	[1; 2]
Desired points in the final map (p)	[100; 40000]	[100; 40000]	[100; 40000]	[100; 40000]	[100; 40000]	[100; 40000]
Total time of image sequence	30	15	30	15	30	15
Time for estimation (u)	Starting at 30. With increments +1.	Starting at 15. With increments +1.	Starting at 30. With increments +1.	Starting at 15. With increments +1.	Starting at 30. With increments +1.	Starting at 15. With increments +1.

Chapter 4

Evaluation

The methodology used for the evaluation of the *Migration* plugin has already been described in the previous chapter. The present chapter will be used to present, analyze and discuss the results obtained with the developed plugin.

The *Migration* plugin was implemented to run in two different ways, according to the availability of the data, i.e., to run with uploaded data files of particles positions previously obtained with other methods or to run calling the *Particle Detector and Tracker* plugin to automatically detect the particles positions used to estimate the probability density functions for the migration of the particles. With this in mind, it is possible to compare the results obtained by the *Migration* plugin taking into account whether the data used is from manual or automatic annotations.

Ultimately, the best results for the estimated particle positions are compared with the actual location of the wound inflicted in the drosophila epithelium.

4.1 Results Obtained

As it was said in the previous chapter, the *Migration* plugin was implemented to run in two different ways, according to the availability of the data, i.e., to run with uploaded data files of particles positions previously obtained with other methods or to run calling the *Particle Detector and Tracker* plugin to automatically detect the particles positions used to estimate the probability density functions for the migration of the particles. With this in mind, it is possible to compare the results obtained by the *Migration* plugin taking into account whether the data used is from manual or automatic annotations.

Ultimately, the best results for the estimated particle positions are compared with the actual location of the wound inflicted in the drosophila epithelium.

4.1.1 Manual Annotation vs. Automatic Annotation

The parameters studied were:

- ◆ a - This value, as described earlier, is associated to the uncertainty ($Q_{tu} = d^{-a}$) of the predicted particles positions. The uncertainty will contribute to the representativity that a predicted position will have in the permanence probability map.
- ◆ *Desired points in the final map* (p) – It represents the total number of particles that will be defined in the final map.
- ◆ *Time for estimation* (u) – This sets the time that will serve to estimate de positions of particles.

In each iteration carried out, the *Migration* plugin was employed with a different combination of parameters (see table 3), allowing the observation of how it would influence the results obtained.

When p is kept constant and all the other parameters vary, it is observed the same pattern for every group of results (see Fig. 10). As the values of a increases, the same happens with the values of the Mean for X and Y (the coordinates of the estimated positions). However, the values regarding the variances are lower as the value a is higher.

If the value a is constant and the other parameters are altered, the calculated means and variances are very close to each other. In fact, in some cases ($a = 5$, see Fig.11), they overlap, which means that the number of the desired points in the representativity map are no longer relevant.

As mentioned in the last chapter, to create the representativity map of each pair particle/frame there will be used the uncertainty (Q , see formula above). The permanence probability map is created after, taking into account the representativity map and the desired number of points in the final map (set by the user). Here, what will influence most dramatically de values of the mean and variance will be the parameter a . Increasing a , and according to the formula of Q , will lend more weight in the particles of the last frames of the image sequence. This is going to increase the number of points being represented that relate to the estimation position from the last frames. As a consequence, the variance decrease, which is a good thing, because it means that there, are not many values different from the mean value. However, using values of a above 5, in spite the low variance, might hide the real distribution of the estimated points – the

points of the initial frames will not be appropriately represented because the uncertainties associated to them will shift towards zero.

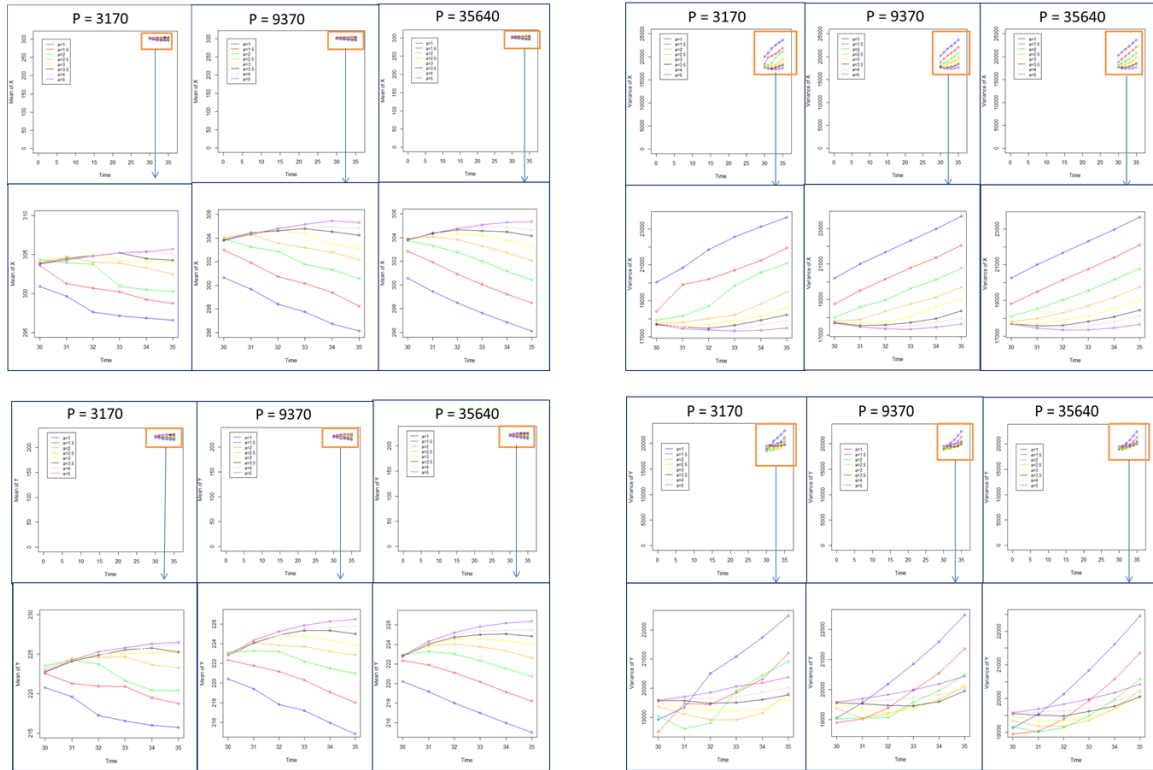


Fig.10. Influence of a (value associated to the uncertainty) in the outcome of the *Migration* plugin. The top-left and bottom-left quadrant characterizes the Mean values and the top-right and bottom-right quadrant represents the values of the Variance (Manual Annotation).

The results presented in Fig.10 and Fig.11 also show the changes of the values over time. Above $a = 2,5$, the graphs of the variances display the existence of a minimum point, which represents a lower dispersion from the mean value. If the value of a is increased, the minimum point appears latter in time (see table 4 and table 5), and the values of the variances are smaller (i.e. the variance decreases).

The first thing that stands out in all results is the fact that the manual annotation only starts giving satisfactory results above $a=2,5$. The automatic annotation has good results at $a = 1$ (see Fig.10 e Fig.12), and presents smaller variances. The reason for this to happen might be in the explanation of Genovesio A et al, that asserts that many biological object motility studies are based on the study of a few hand-picked particles which represent only a small subset of the total (rarely all particles in an image sequence are considered) [15]. If the particles of manual annotations represent only a small subset of the total, and the automatic annotations represent the total, it is logical to

have lower variances, because as higher the sample size, lower will be the variability shown by the estimator mean (and the variance decreases).

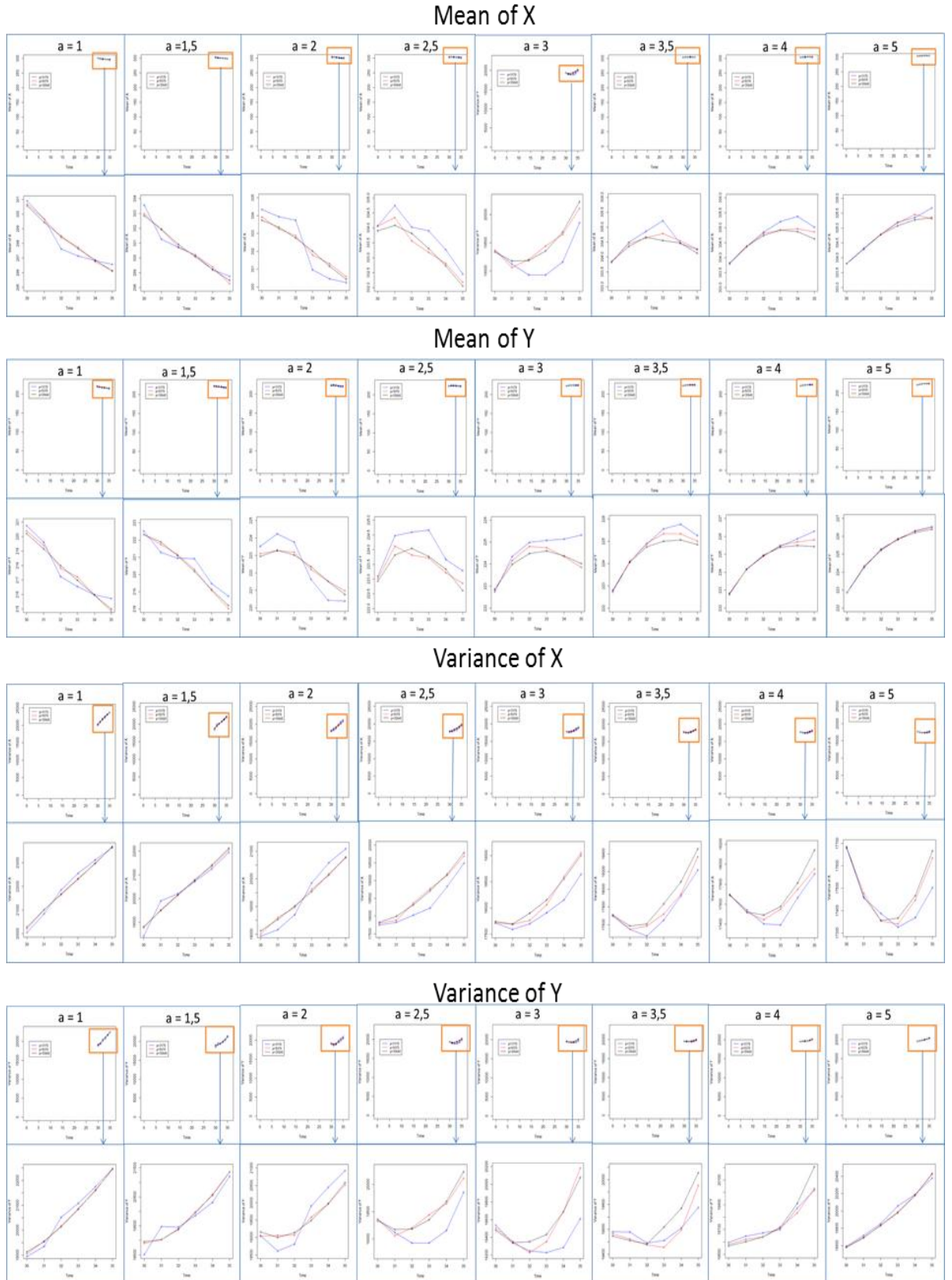


Fig. 10 Influence of p (desired points in the final map) in the outcome of the *Migration* plugin.

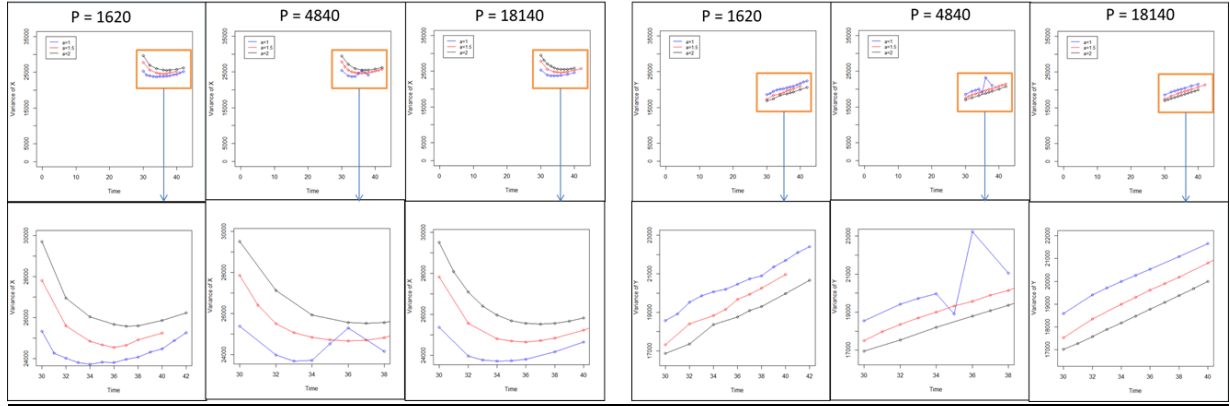


Fig. 12 Influence of a (value associated to the uncertainty) in the outcome of the *Migration* plugin. The results from the left characterizes the Variance of X and on the right represents the values of the Variance of Y (Automatic Annotation).

Table 4. Best results of the *Migration* plugin from the image sequence
;SrpHemoGal4UASnlsCherryUbiCad;after_proj.tiff (FilmB) – Manual Annotation.

p	a	u	Mean X	Mean Y	Variance of X	Variance of Y
3170	2	31	303,9442	224,2456	18161	18611,2
3170	2,5	33	303,9063	224,6652	18220,88	18924,51
3170	3	31	304,5421	224,3454	17593,24	19336,86
3170	3,5	32	304,8491	224,8584	17474,2	19489,56
3170	4	33	305,2078	225,4511	17392,51	19698,26
3170	5	33	305,2022	225,8173	17325,94	20057,49
9370	2,5	31	304,3465	224,122	17877,68	19059,37
9370	3	31	304,2904	224,1568	17689,02	19354,29
9370	3,5	31	304,4412	224,0828	17547,16	19522,93
9370	4	32	304,8127	224,889	17443,75	19640,96
9370	5	33	305,1721	225,8481	17339,66	19987,22
35640	2	31	303,3409	223,2862	18535,56	19004,29
35640	2,5	31	304,0876	223,819	17993,77	19174,1
35640	3	31	304,3572	223,9817	17692,73	19340,53
35640	3,5	31	304,3505	224,0519	17585,59	19510,03
35640	4	32	304,7443	224,9506	17489,91	19640,09
35640	5	32	304,7764	225,2111	17356,49	19835,99

Table 5. Best results of the *Migration* plugin from the image sequence
;SrpHemoGal4UASnlsCherryUbiCad;after_proj.tiff (FilmB) – Automatic Annotation.

<i>p</i>	<i>a</i>	<i>u</i>	Mean X	Mean Y	Variance of X	Variance of Y
1620	1	34	294,3796	263,1292	23714,43	20059,78
1620	1,5	36	297,7935	262,0078	24541,97	19665,76
1620	2	37	299,5709	262,497	25573,38	19102,75
4840	1	33	294,4989	263,339	23693,91	19730,39
4840	1,5	36	297,3352	262,1062	24670,91	19579,14
4840	2	37	299,6058	261,9735	25527,52	19085,84
18140	1	34	294,0991	262,8382	23715,52	19989,11
18140	1,5	36	297,3892	262,23	24658,64	19604,63
18140	2	37	300,1987	262,0646	25525,04	19068,42

Table 6. Best results of the *Migration* plugin from the image sequence
SrpHemoGal4UASnlsCherryUbiCadGFPaft_proj.tiff (Film A)) – Manual Annotation.

<i>p</i>	<i>a</i>	<i>u</i>	Mean X	Mean Y	Variance of X	Variance of Y
990	2,5	17	203,9276	242,1631	14288,33	14025,63
990	3	18	204,5562	240,9909	14216,15	14226,43
990	3,5	19	205,2259	240,1378	14205,42	14557,33
990	4	17	199,7378	246,9571	14210,45	13705,03
990	5	17	198,7297	248,3306	14224,18	13706,48
2330	2,5	16	201,7466	244,1712	14721,73	13804,77
2330	3	16	199,9372	246,5183	14457,65	13623,72
2330	3,5	17	200,699	245,5803	14264,82	13813,14
2330	4	18	202,3577	244,0376	14255,45	14098,49
2330	5	19	201,8892	244,6248	14043,90	14365,33
6510	2,5	16	201,3188	244,2586	14717,59	13816,6
6510	3	16	200,097	246,096	14606,82	13676,95
6510	3,5	17	201,2047	244,9101	14470,27	13895,53
6510	4	17	200,364	246,2832	14330,57	13809,34
6510	5	18	200,5776	246,3848	14124,87	14013,13

Table 7. Best results of the *Migration* plugin from the image sequence
SrpHemoGal4UASnlsCherryUbiCadGFPaft_proj.tiff (Film A)) – Automatic Annotation.

<i>p</i>	<i>a</i>	<i>u</i>	Mean X	Mean Y	Variance of X	Variance of Y
790	2	20	193,5857	252,2873	4610,047	7539,184
790	1,5	22	191,9179	252,2019	4533,832	7693,684
790	2	26	188,3235	253,5758	4634,594	8023,305
1840	1	20	191,4121	252,3999	4629,27	7817,352
1840	1,5	23	191,4121	252,3999	4629,27	7817,352
1840	2	24	190,0516	252,5105	4698,043	7686,066
5040	1	21	192,6402	252,4571	4612,559	7703,121
5040	1,5	22	191,7634	252,2966	4671,824	7634,566
5040	2	24	190,066	252,5286	4704,223	7703,512

4.1.2 Comparison with the wound location

In this section, the primer question to answer is: does all the estimated location of the particles tend to concentrate in the wound sites of the *drosophila* epithelium?

Table 8. Coordinates and area of the wound location.

Image Sequence	X (center of Mass)	Y (center of mass)	Area
SrpHemoGal4UASnlsCherryUbi iCadGFPaft_proj.tiff (Film A)	269,31	222,92	2046,47
;SrpHemoGal4UASnlsCherryUbi Cad;after_proj.tiff (FilmB)	280,43	257,02	15166

From the analysis of the results (see tables 4, 5, 6 and7), and comparing them to values of the wound (Table 8), some of the positions estimated are no so far from the actual location of the inflicted wound. The wound is not just the x and y location presented, it also has a certain area, so the differences presented can are contained in the area of the wound. In other words, the mean is the measure of the center of the distribution. The variance is the arithmetic average of the squared differences between the values and the mean (the units of variance are the square of the physical unit of the data. The variance is a descriptor of a probability distribution, describing how widely a set of points varies from the mean (i.e., expected value) of the points, regarding an area.

Table 9. Distances of the mean values to the center of mass of the wound.

		Mean Value		Distance	
		Film A	Film B	Film A	Film B
Manual	X min	198,7297	303,3409	-70,5803	22,9109
	X max	205,2259	305,2078	-64,0571	24,7778
	Ymin	240,1378	223,2862	17,2178	-57,1438
	Ymax	248,3306	225,8481	25,4106	-54,5819
Automático	X min	188,3235	294,0991	-80,9865	13,6691
	X max	193,5857	300,1987	-75,7243	19,7687
	Ymin	252,2019	261,9735	29,2819	-18,4565
	Ymax	225,8481	263,339	30,6558	-17,091

4.2 Applicability and Efficiency

The plugin developed was written in the programming language Java and is a simple algorithm. It can be used to process image sequences to extract information regarding the estimation of the future positions of particles in a determined time point in the future.

The user interface was designed to achieve three major goals. First, user friendliness was wanted for fast and intuitive experimental design. All command and parameter settings required to execute the plugin are accessible from a single window. The second goal was flexibility so that the user can use all values wanted and the plugin can be cancelled at any time during the processing. The third goal was speed and robustness, to minimize the time required to end the algorithm.

Regarding efficiency, the algorithm executes quickly, only using fractions of a second.

4.3 Consistency and Completeness

The plugin has consistency because, as it can be seen from the results, despite of using different points for the final map, the variances have almost the same values for the same results of values associated to uncertainty.

This method also allows completeness because, using the automated particle tracker method to detect the particles, all the particles in each frame are used for de subsequent analysis.

Chapter 5

Conclusions

The evaluation done to the *Migration* plugin does not include treatment and analysis of the physical system under observation and the used imaging equipment.

This new method reasoned that an accurate representation of the movement characteristics of trafficked particles could be obtained without the need for continuous tracking of individual motile particles over their entire path, i.e. the only values needed are the particles position detected at a certain time point.

Comparing the present solution to the existing approaches, this method adds an improvement. A fixed number of objects found in every frame of the image sequence are an impractical assumption. In microscopy, new objects appear by their coming to the field of view. At the same time some existing object can disappear by moving out of focus, changing identity, or fusing with another object [20]. The present approach can deal with complex situation, such as overlapping of particles or other occlusions. The reasons are related to the only need for knowing the particles position at a time point.

This approach is based on a simple analysis of the cell paths between two frames and allows the determination at a glance, of the distribution of cells in a determined time point (set by the user). The premise is: knowing the behavior in the past is possible to know the behavior of particles in the future (not forgetting that a value of uncertainty is always added to the prediction made).

This approach is capable to use data that has to be immediately extracted from an image sequence and being capable to handle information, saved in a text file, formerly extracted from an image sequence. This characteristic adds flexibility in the use of the plugin.

From the analysis of the results, the *Migration* plugin presents satisfactory results. The estimated distributions tend to be very close to the wound inflicted in the fly, in fact higher probabilities are concentrated above the wound. The mean of the distribution is

not equal to the center of mass of the wound, but the distances calculated show that is located in the area of the wound. The technique presented, works with both manual annotations and automatic annotations, although the automatic annotation has better results. The characteristics of the images that were analyzed were analogous, and so, the analysis of this plugin should be extended to other images (with different acquisition conditions and different object characteristics) to see if there are changes in the results obtained and if the method is applicable. There is no other similar method that can be used as a comparison to this approach.

With the development of more robust automated methods for analyzing cell migration it will become more and more possible to accumulate large amounts of information databases that allow statistical distinction of behavioral heterogeneity. The analysis of behavioral heterogeneity defines an emerging paradigm in molecular biology, the goal of which is to identify all possible states and the relevant states transitions of a system in its natural mode of action [23]. This approach is likely to reveal data that allows the comprehension of the mechanism of cellular homeostasis that underlies robustness in life. Knowing the state of a healthy molecular/cellular process, it will be much easier to understand abnormal behavior that leads to disease and to define strategies that return the deviated system to its normal states.

References

1. Acton ST, Wethmar K, Ley K. Automatic tracking of Rolling Leukocytes in Vivo. *Microvasc Res*. 2002; 63(1): 139-48.
2. Ananthakrishnan R, Ehrlicher A. The Forces Behind Cell Movement. *Int J Biol Sci*. 2007; 3(5): 303-17.
3. Andrews S, Gilley J, Coleman MP. Difference Tracker: ImageJ plugins for fully automated analysis of multiple axonal transport parameters. *J Neurosci Methods*. 2010; 192(2): 281-7.
4. Anthony SM, Kim M, Granick S. Single-particle tracking of janus colloids in close proximity. *Langmuir*. 2008; 24: 6557-61.
5. Ariano P, Distasi C, Gilardino A, Zamburlin P, Ferraro M. A simple method to study cellular migration. *J Neurosci Methods*. 2004; 141(2): 271-6.
6. Baker RE, Yates CA, Erban R. From microscopic to macroscopic descriptions of cell migration on growing domains. *Bull Math Biol*. 2010; 72(3): 719-62.
7. Beltman JB, Marée AFM, Boer RJ. Analysing immune cell migration. *Nat Rev Immunol*. 2009; 9(11): 789-98.
8. Biwole PH, Favier E, Krauss G, Roux JJ, Rusaouen G. Development of a high speed computer vision system for 3D particle tracking in full scale rooms. 3rd International Conference on Information and Communication Technologies From Theory to Applications; 2008 April 7-11; Damascus. *ICTTA 2008*; 15: 1-6.
9. Bonnet N, Delavoie F, Zahm JM. Characterizing the spatio-temporal behavior of cell populations through image auto- and cross-correlation microscopy. *Biotechniques*. 2007; 43(1): 107-15.
10. Burger W, BurgeMJ. Digital image processing: an algorithmic introduction using Java. 1st ed. New York: Springer; 2008.

11. Cheezum MK, Walker WF, Guilford WH. Quantitative comparison of algorithms for tracking single fluorescent particles. *Biophys J*. 2001; 81(4): 2378-88.
12. Chertkov M, Kroc L, Krzakala F, Vergassola M, Zdeborová L. Inference in particle tracking experiments by passing messages between images. *Proc Natl Acad Sci USA*. 2010; 107(17): 7663-8.
13. Cordelières FP. Manual Tracking. Institut Curie, Orsay (France); 2005 [accessed in 2010 October 17]. Available in: <http://rsbweb.nih.gov/ij/plugins/track/Manual%20Tracking%20plugin.pdf>
14. Ferreira T, Rasband W. ImageJ user Guide. U. S. National Institutes of Health, MD; 2011 [accessed in 2011 August 12]. Available in: <http://rsbweb.nih.gov/ij/docs/user-guide.pdf>
15. Genovesio A, Liedl T, Emiliani V, Parak WJ, Coppey-Moisand M, Olivio-Marin JC. Multiple particle tracking in 3-D+t microscopy: method and application to the tracking of endocytosed quantum dots. *IEEE Trans Image Process*. 2006; 15(5): 1062-70.
16. Gerlich D, Mattes J, Eils R. Quantitative motion analysis and visualization of cellular structures. *Methods*. 2003; 29(1): 3-13.
17. Gustafsson F, Gunnarsson F, Bergman N, Forssell U, Jansson J, Karlsson R, Nordlund PJ. Particle filters for positioning, navigation and tracking. *IEEE Trans Signal Process*. 2002; 50(2): 425-37.
18. Huth J, Buchholz M, Kraus JM, Schmucker M, von Wichert G, Krndija D, Seufferlein T, Gress TM, Kestler HA. Significantly improved precision of cell migration analysis in time-lapse video microscopy through use of a fully automated tracking system. *BMC Cell Biol*. 2010; 11: 1-12.
19. Kalaidzidis Y. Multiple objects tracking in fluorescence microscopy. *J Math Biol*. 2009; 58: 57-80.
20. Kalaidzidis Y. Intracellular objects tracking. *Eur J Cell Biol*. 2007; 86: 569-78.
21. Kleinfelter N, Moroni M, Cushman JH. Application of the finite-size Lyapunov exponent to particle tracking velocimetry in fluid mechanics experiments. *Phys Rev E Stat Nonlin Soft Matter Phys*. 2005; 72(5 Pt 2): 056306.

22. Levy G. Particle detector and tracker. MOSAIC group, ETH Zurich; 2005 [accessed in 2010 October 21]. Available in: <https://weeman.inf.ethz.ch/ParticleTracker/#general>
23. Meijering E, Smal I, Danuser G. Tracking in molecular bioimaging. *IEEE Signal Process Mag.* 2006; 23(3): 46-53.
24. Miura K. Tracking movement in cell biology. *Adv Biochem Eng Biotechnol.* 2005; 95: 267-95.
25. Rabut G, Ellenberg J. Automatic real-time three-dimensional cell tracking by fluorescence microscopy. *J Microsc.* 2004; 216(Pt 2): 131-7.
26. Sage D, Neumann FR, Hediger F, Gasser SM, Unser M. Automatic tracking of individual fluorescence particles: application to the study of chromosome dynamics. *IEEE Trans Image Process.* 2005; 14(9): 1372-83.
27. Sbalzarini IF, Koumoutsakos P. Feature point tracking and trajectory analysis for video imaging in cell biology. *J Struct Biol.* 2005; 151(2): 182–95.
28. Smal I, Meijering E, Draegestein K, Galjart N, Grigoriev I, Akhmanova A, van Royen ME, Houtsmuller AB, Niessen W. Multiple object tracking in molecular bioimaging by Rao-blackwellized marginal particle filtering. *Med Image Anal.* 2008; 12: 764-77.
29. Stefanoni F, Ventre M, Mollica F, Netti P. A numerical model for durotaxis. *J Theor Biol.* 2011; 280(1): 150-8.
30. Villa CE, Caccia M, Sironi L, D'Alfonso L, Collini M, Rivolta I, Miserocchi G, Gorletta T, Zanoni I, Granucci F, Chirico G. Accumulative difference image protocol for particle tracking in fluorescence microscopy tested in mouse lymphonodes. *PLoS One.* 2010; 5(8): e12216.
31. Zimmer C, Labruyère E, Meas-Yedid V, Guillén N, Olivio-Marin JC. Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: a tool for cell-based drug testing. *IEEE Trans Med Imaging.* 2002; 21(10): 1212-21.
32. Vasquez D, Fiaichaxd T. Motion Prediction for Moving Objects: a Statistical Approach. *Proc IEEE Int Conf Robot Autom.* 2004.
33. Jeung H, Liu Q, Shen H, Zhou X. A Hybrid Prediction Model for Moving Objects. *Proc Int Conf Biomed Eng Inform.* 2008.